DOCUMENTATION

Programming Project

subgroup: 22

cluster: 6.2

Students:

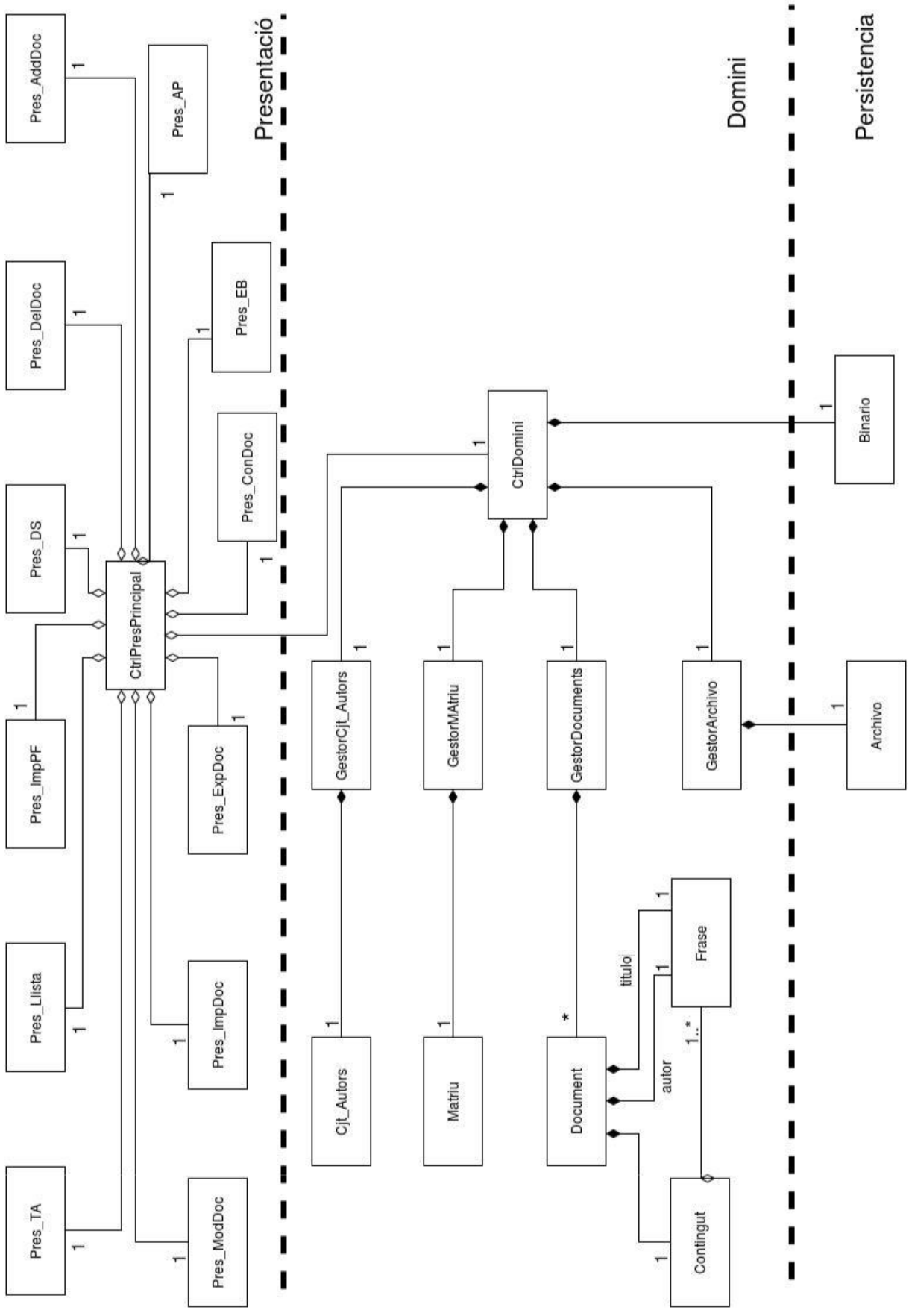
Albert Ruiz Lombarte

Aleix Rosello Rosiñol

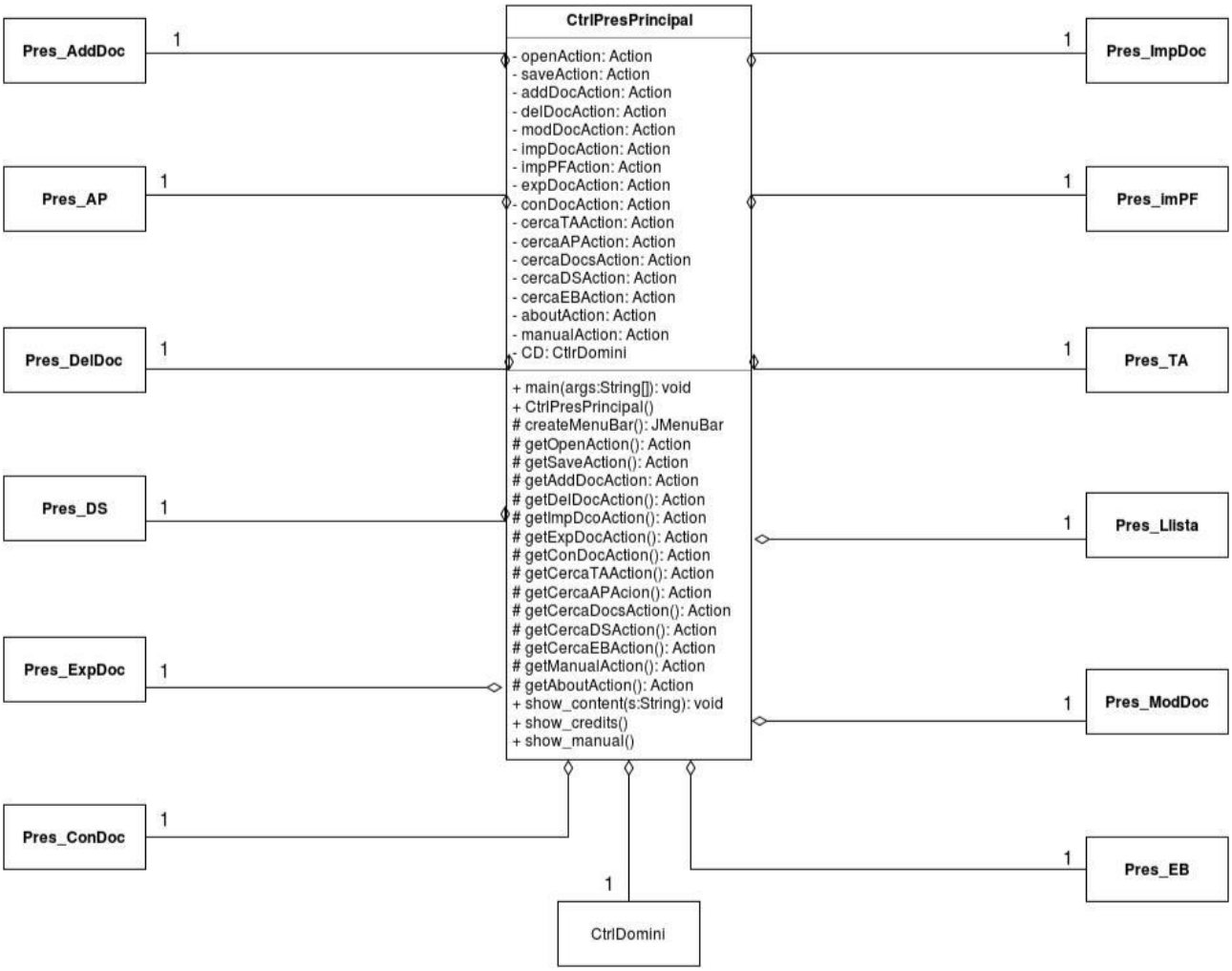
John Bryan Game Glices

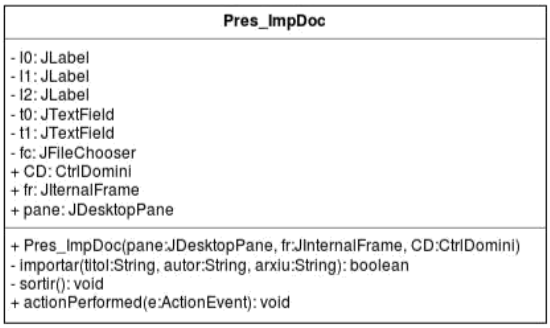
index

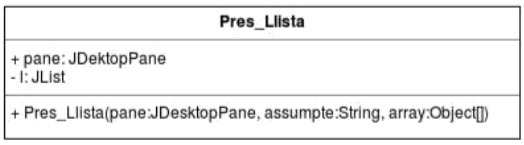
1. Design of the three-tier architecture (MVC)

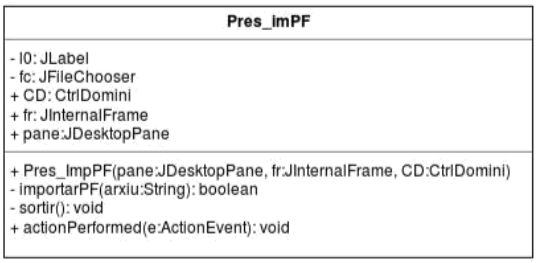


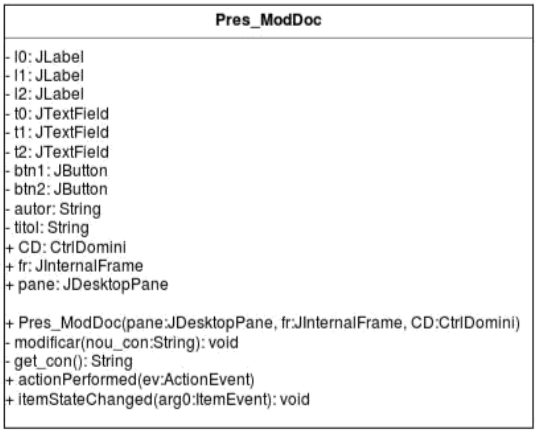
2. Complete static UML diagrams generated separately

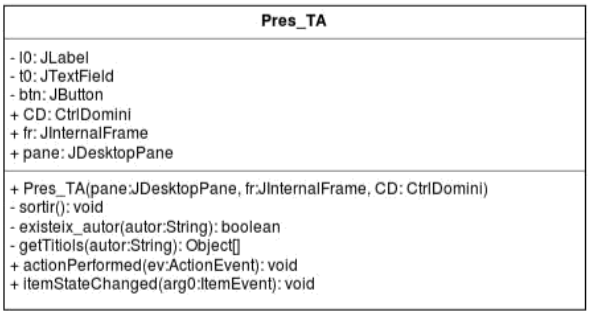


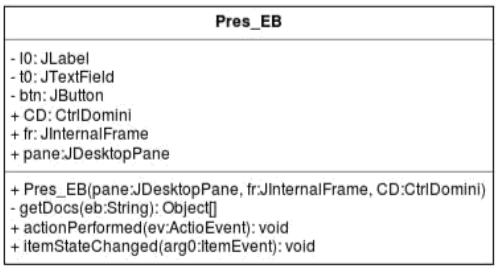

Pres_ImpDoc
- l0: JLabel
- l1: JLabel
- l2: JLabel
- t0: JTextField
- t1: JTextField
- fc: JFileChooser
+ CD: CtrlDomini
+ fr: JInternalFrame
+ pane: JDesktopPane
+ Pres_ImpDoc(pane:JDesktopPane, fr:JInternalFrame, CD:CtrlDomini)
- importar(titol:String, autor:String, arxiu:String): boolean
- sortir(): void
+ actionPerformed(e:ActionEvent): void

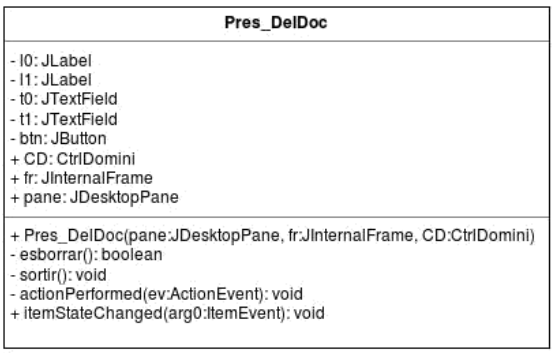

Pres_Llista
+ pane: JDektopPane
- l: JList
+ Pres_Llista(pane:JDesktopPane, assumpte:String, array:Object[])

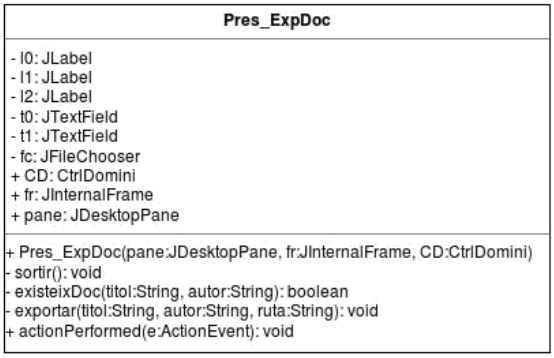

Pres_ImPF
- l0: JLabel
- fc: JFileChooser
+ CD: CtrlDomini
+ fr: JInternalFrame
+ pane:JDesktopPane
+ Pres_ImpPF(pane:JDesktopPane, fr:JInternalFrame, CD:CtrlDomini)
- importarPF(arxiu:String): boolean
- sortir(): void
+ actionPerformed(e:ActionEvent): void

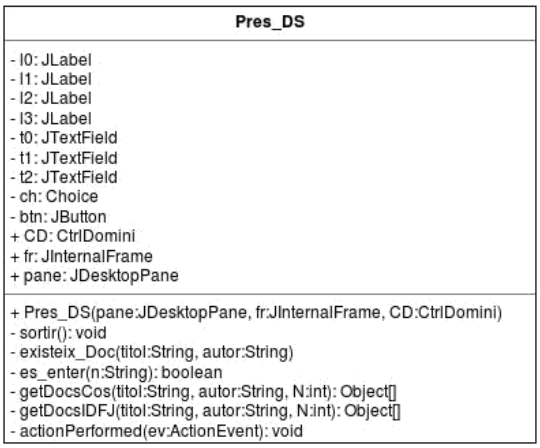

Pres_ModDoc
- l0: JLabel
- l1: JLabel
- l2: JLabel
- t0: JTextField
- t1: JTextField
- t2: JTextField
- btn1: JButton
- btn2: JButton
- autor: String
- titol: String
+ CD: CtrlDomini
+ fr: JInternalFrame
+ pane: JDesktopPane
+ Pres_ModDoc(pane:JDesktopPane, fr:JInternalFrame, CD:CtrlDomini)
- modificar(nou_con:String): void
- get_con(): String
+ actionPerformed(ev:ActionEvent)
+ itemStateChanged(arg0:ItemEvent): void

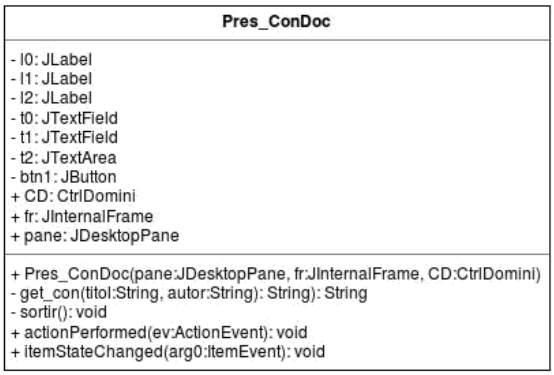

Pres_TA
- l0: JLabel
- t0: JTextField
- btn: JButton
+ CD: CtrlDomini
+ fr: JInternalFrame
+ pane: JDesktopPane
+ Pres_TA(pane:JDesktopPane, fr:JInternalFrame, CD: CtrlDomini)
- sortir(): void
- existeix_autor(autor:String): boolean
- getTitols(autor:String): Object[]
+ actionPerformed(ev:ActionEvent): void
+ itemStateChanged(arg0:ItemEvent): void

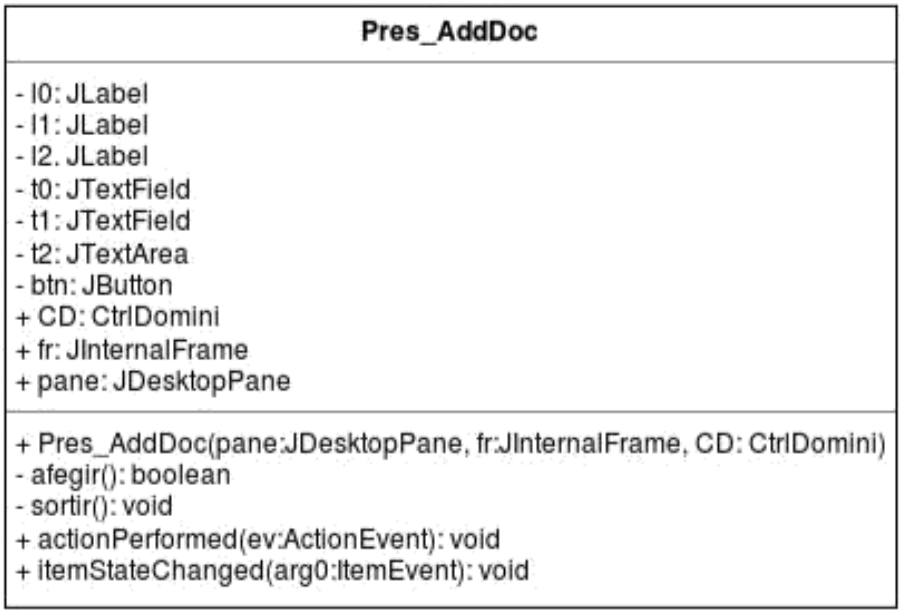

Pres_EB
- l0: JLabel
- t0: JTextField
- btn: JButton
+ CD: CtrlDomini
+ fr: JInternalFrame
+ pane:JDesktopPane
+ Pres_EB(pane:JDesktopPane, fr:JInternalFrame, CD:CtrlDomini)
- getDocs(eb:String): Object[]
+ actionPerformed(ev:ActioEvent): void
+ itemStateChanged(arg0:ItemEvent): void

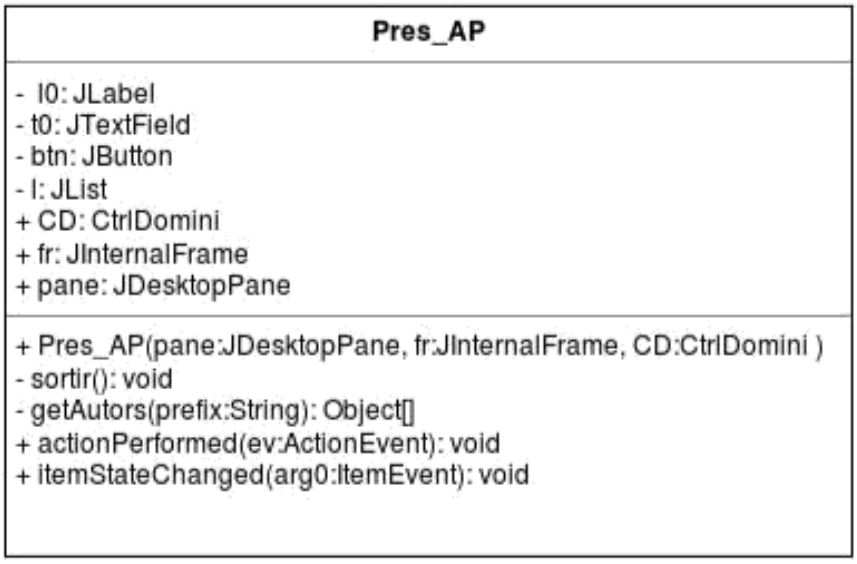

Pres_DelDoc
- l0: JLabel
- l1: JLabel
- t0: JTextField
- t1: JTextField
- btn: JButton
+ CD: CtrlDomini
+ fr: JInternalFrame
+ pane: JDesktopPane
+ Pres_DelDoc(pane:JDesktopPane, fr:JInternalFrame, CD:CtrlDomini)
- esborrar(): boolean
- sortir(): void
- actionPerformed(ev:ActionEvent): void
+ itemStateChanged(arg0:ItemEvent): void

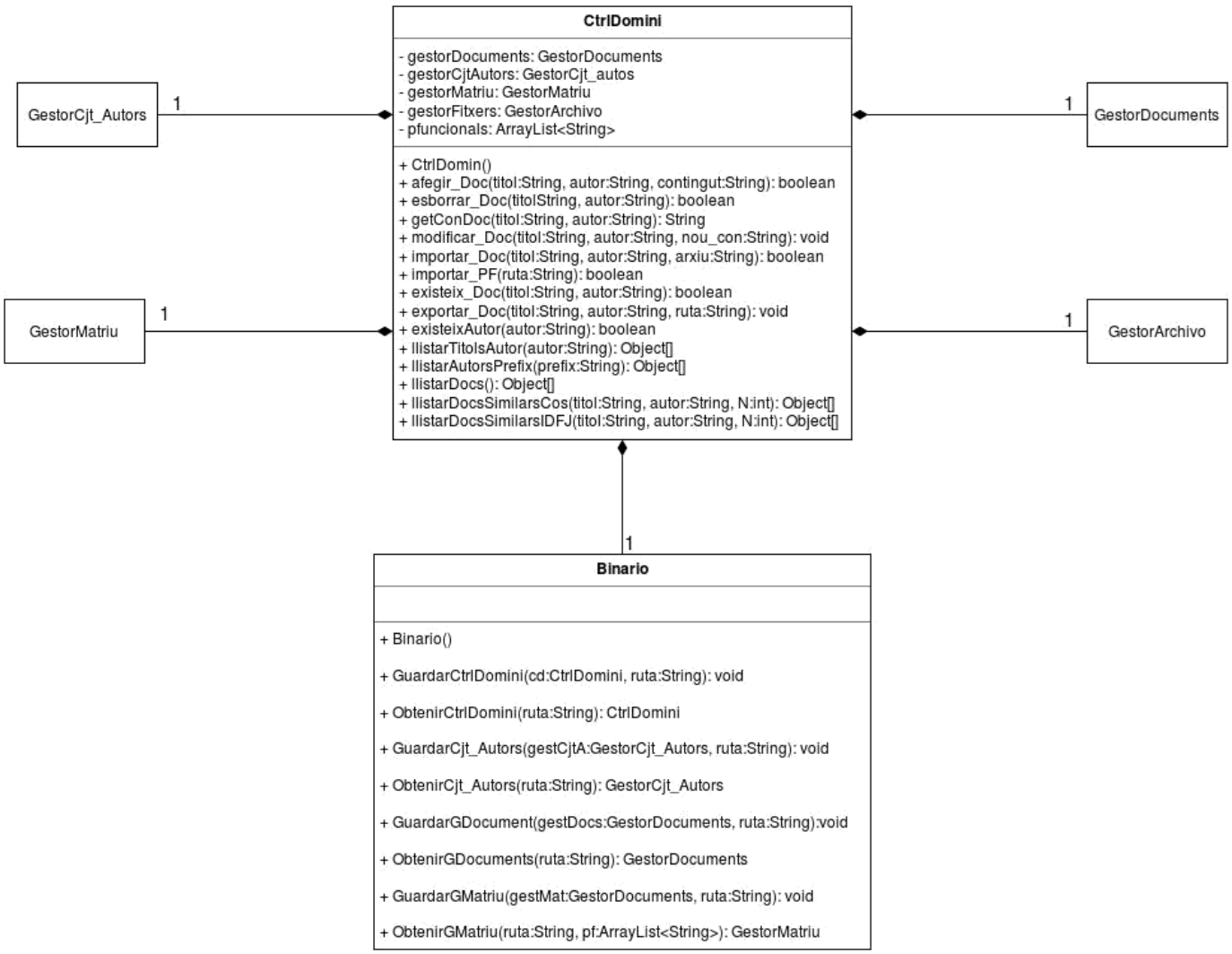

Pres_ExpDoc
- l0: JLabel
- l1: JLabel
- l2: JLabel
- t0: JTextField
- t1: JTextField
- fc: JFileChooser
+ CD: CtrlDomini
+ fr: JInternalFrame
+ pane: JDesktopPane
+ Pres_ExpDoc(pane:JDesktopPane, fr:JInternalFrame, CD:CtrlDomini)
- sortir(): void
- existeixDoc(titol:String, autor:String): boolean
- exportar(titol:String, autor:String, ruta:String): void
+ actionPerformed(e:ActionEvent): void

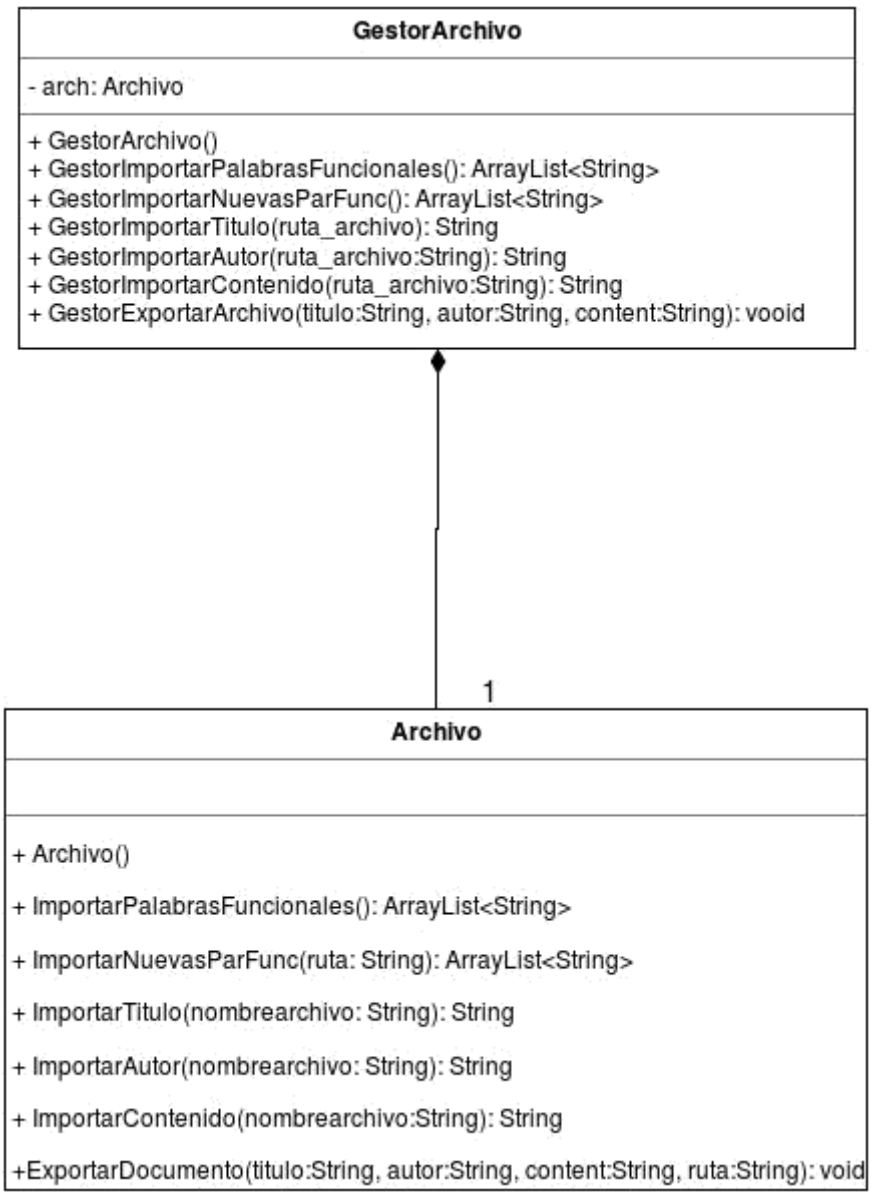

Pres_DS
- l0: JLabel
- l1: JLabel
- l2: JLabel
- l3: JLabel
- t0: JTextField
- t1: JTextField
- t2: JTextField
- ch: Choice
- btn: JButton
+ CD: CtrlDomini
+ fr: JInternalFrame
+ pane: JDesktopPane
+ Pres_DS(pane:JDesktopPane, fr:JInternalFrame, CD:CtrlDomini)
- sortir(): void
- existeix_Doc(titol:String, autor:String)
- es_enter(n:String): boolean
- getDocsCos(titol:String, autor:String, N:int): Object[]
- getDocsIDFJ(titol:String, autor:String, N:int): Object[]
- actionPerformed(ev:ActionEvent): void

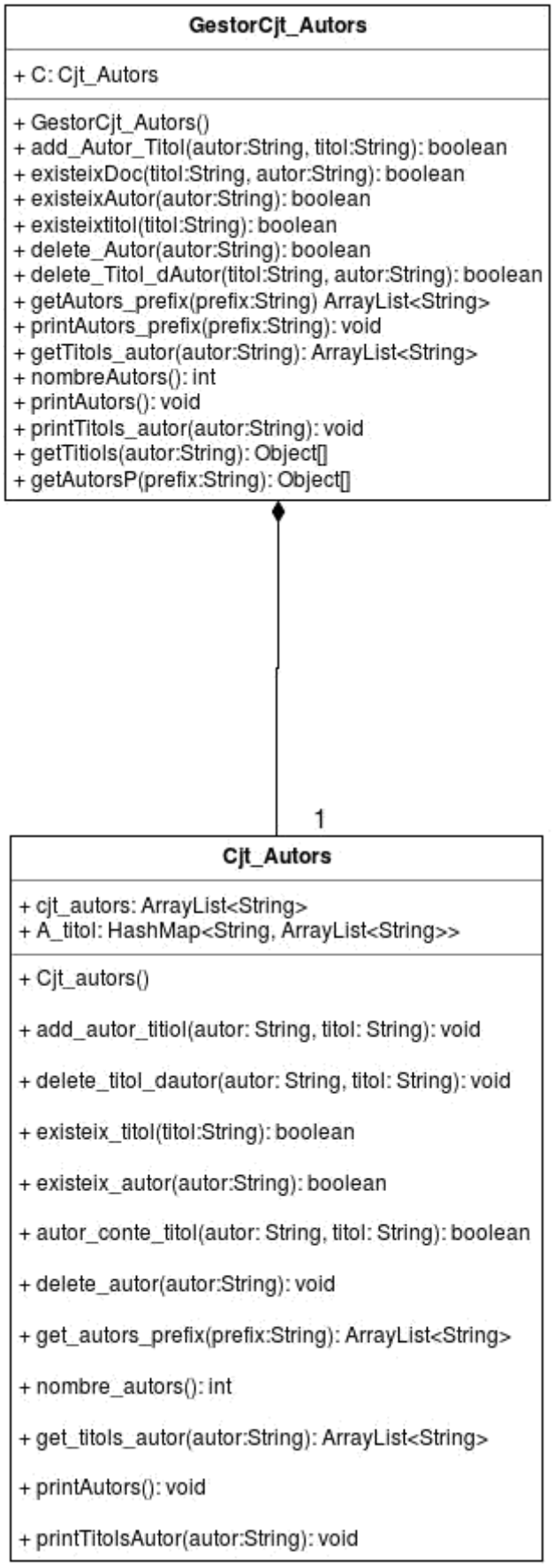

Pres_ConDoc
- l0: JLabel
- l1: JLabel
- l2: JLabel
- t0: JTextField
- t1: JTextField
- t2: JTextArea
- btn1: JButton
+ CD: CtrlDomini
+ fr: JInternalFrame
+ pane: JDesktopPane
+ Pres_ConDoc(pane:JDesktopPane, fr:JInternalFrame, CD:CtrlDomini)
- get_con(titol:String, autor:String): String): String
- sortir(): void
+ actionPerformed(ev:ActionEvent): void
+ itemStateChanged(arg0:ItemEvent): void

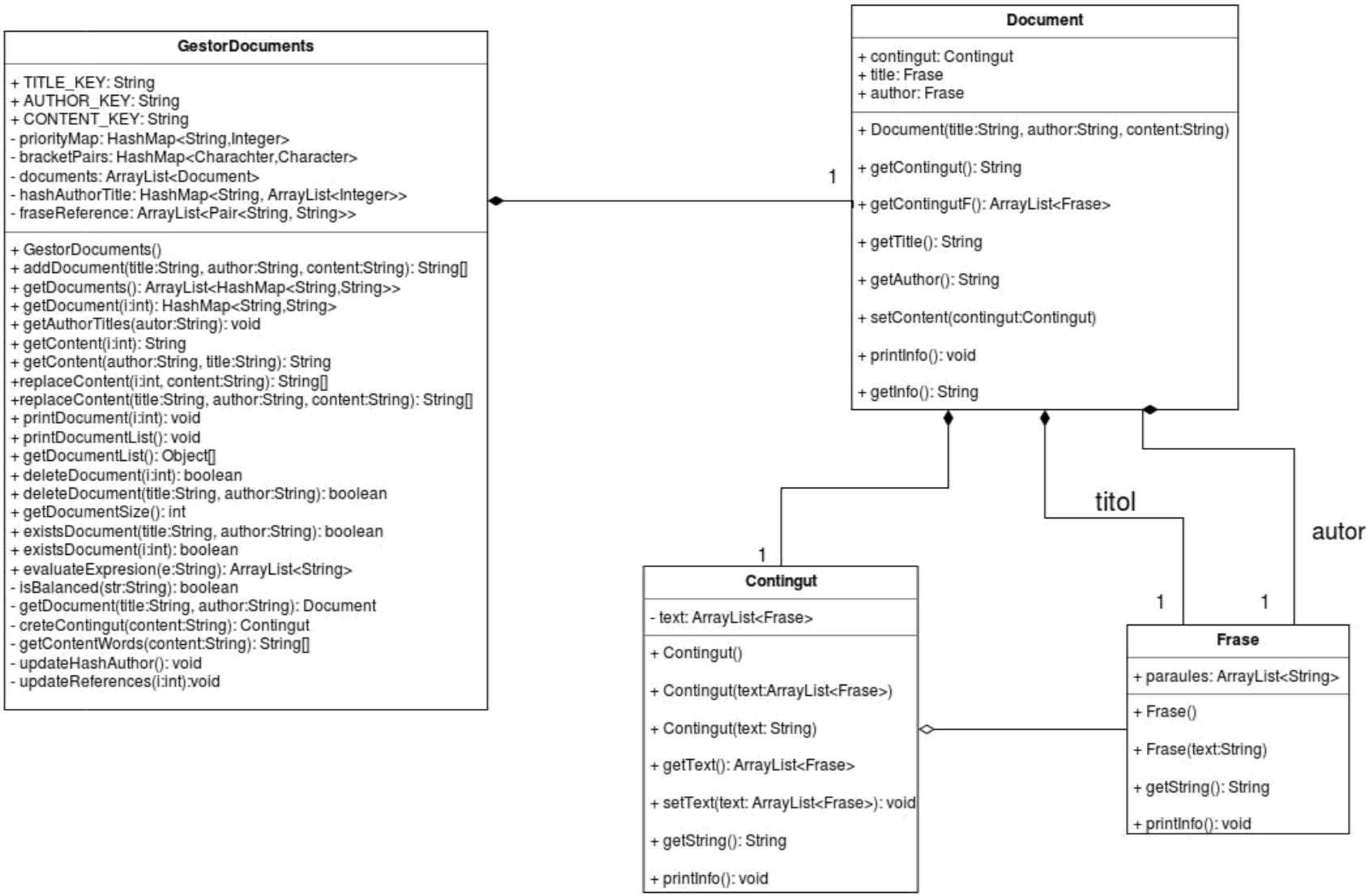

Pres_AddDoc
- l0: JLabel
- l1: JLabel
- l2. JLabel
- t0: JTextField
- t1: JTextField
- t2: JTextArea
- btn: JButton
+ CD: CtrlDomini
+ fr: JInternalFrame
+ pane: JDesktopPane
+ Pres_AddDoc(pane:JDesktopPane, fr:JInternalFrame, CD: CtrlDomini)
- afegir(): boolean
- sortir(): void
+ actionPerformed(ev:ActionEvent): void
+ itemStateChanged(arg0:ItemEvent): void

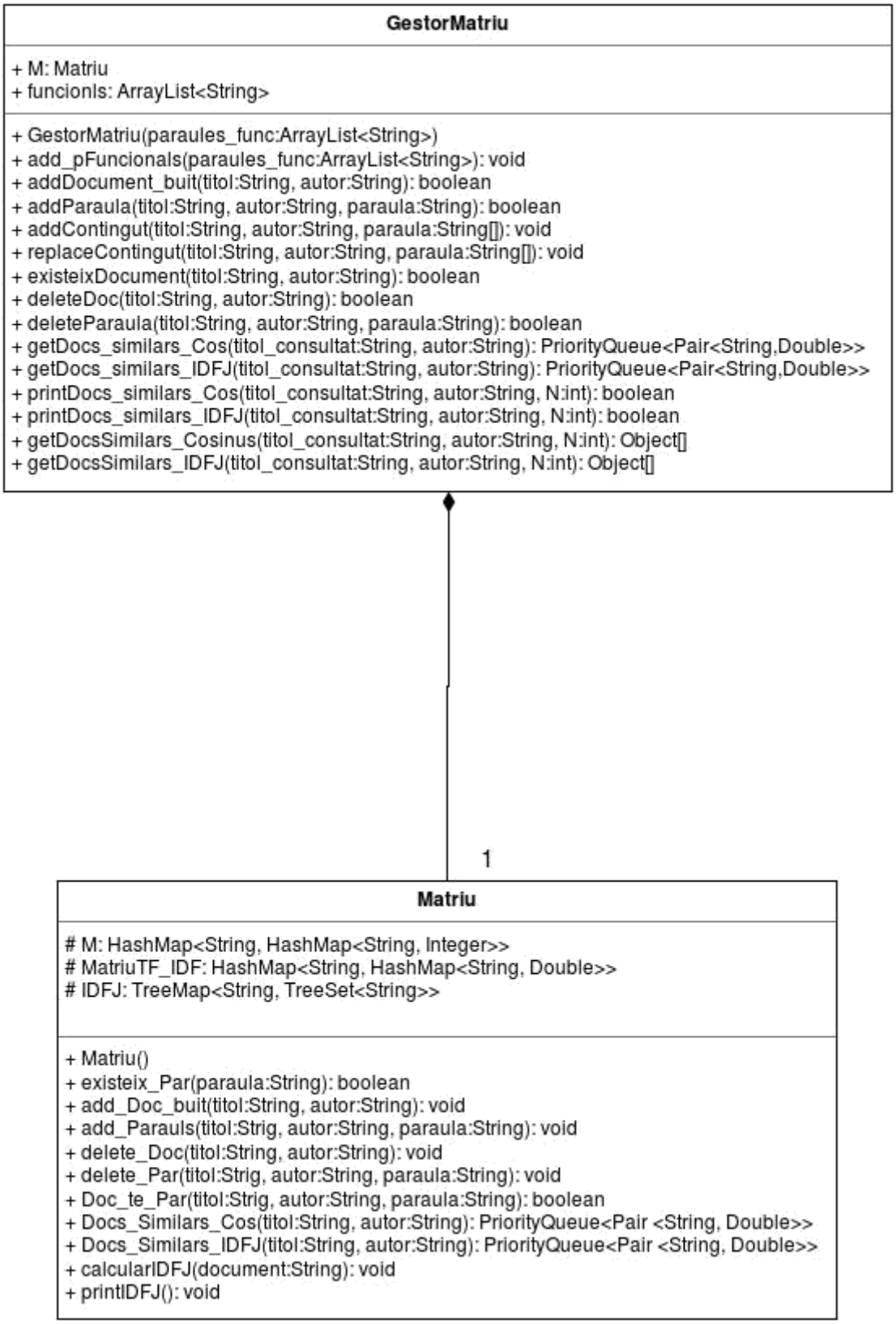

Pres_AP
- l0: JLabel
- t0: JTextField
- btn: JButton
- l: JList
+ CD: CtrlDomini
+ fr: JInternalFrame
+ pane: JDesktopPane
+ Pres_AP(pane:JDesktopPane, fr:JInternalFrame, CD:CtrlDomini)
- sortir(): void
- getAutors(prefix:String): Object[]
+ actionPerformed(ev:ActionEvent): void
+ itemStateChanged(arg0:ItemEvent): void


CtrlDomini
- gestorDocuments: GestorDocuments
- gestorCjtAutors: GestorCjt_autos
- gestorMatriu: GestorMatriu
- gestorFitxers: GestorArchivo
- pfuncionals: ArrayList<String>
+ CtrlDomin()
+ afegir_Doc(titol:String, autor:String, contingut:String): boolean
+ esborrar_Doc(titolString, autor:String): boolean
+ getConDoc(titol:String, autor:String): String
+ modificar_Doc(titol:String, autor:String, nou_con:String): void
+ importar_Doc(titol:String, autor:String, arxiu:String): boolean
+ importar_PF(ruta:String): boolean
+ existeix_Doc(titol:String, autor:String): boolean
+ exportar_Doc(titol:String, autor:String, ruta:String): void
+ existeixAutor(autor:String): boolean
+ llistarTitolsAutor(autor:String): Object[]
+ llistarAutorsPrefix(prefix:String): Object[]
+ llistarDocs(): Object[]
+ llistarDocsSimilarsCos(titol:String, autor:String, N:int): Object[]
+ llistarDocsSimilarsIDFJ(titol:String, autor:String, N:int): Object[]
GestorCjt_Autors
GestorDocuments
GestorMatriu
GestorArchivo
1
1
1
1
1
Binario
+ Binario()
+ GuardarCtrlDomini(cd:CtrlDomini, ruta:String): void
+ ObtenirCtrlDomini(ruta:String): CtrlDomini
+ GuardarCjt_Autors(gestCjtA:GestorCjt_Autors, ruta:String): void
+ ObtenirCjt_Autors(ruta:String): GestorCjt_Autors
+ GuardarGDocument(gestDocs:GestorDocuments, ruta:String):void
+ ObtenirGDocuments(ruta:String): GestorDocuments
+ GuardarGMatriu(gestMat:GestorDocuments, ruta:String): void
+ ObtenirGMatriu(ruta:String, pf:ArrayList<String>): GestorMatriu


GestorArchivo
- arch: Archivo
+ GestorArchivo()
+ GestorImportarPalabrasFuncionales(): ArrayList<String>
+ GestorImportarNuevasParFunc(): ArrayList<String>
+ GestorImportarTitulo(ruta_archivo): String
+ GestorImportarAutor(ruta_archivo:String): String
+ GestorImportarContenido(ruta_archivo:String): String
+ GestorExportarArchivo(titulo:String, autor:String, content:String): vooid
1
Archivo
+ Archivo()
+ ImportarPalabrasFuncionales(): ArrayList<String>
+ ImportarNuevasParFunc(ruta: String): ArrayList<String>
+ ImportarTitulo(nombrearchivo: String): String
+ ImportarAutor(nombrearchivo: String): String
+ ImportarContenido(nombrearchivo:String): String
+ExportarDocumento(titulo:String, autor:String, content:String, ruta:String): void


GestorCjt_Autors
+ C: Cjt_Autors
+ GestorCjt_Autors()
+ add_Autor_Titol(autor:String, titol:String): boolean
+ existeixDoc(titol:String, autor:String): boolean
+ existeixAutor(autor:String): boolean
+ existeixtitol(titol:String): boolean
+ delete_Autor(autor:String): boolean
+ delete_Titol_dAutor(titol:String, autor:String): boolean
+ getAutors_prefix(prefix:String) ArrayList<String>
+ printAutors_prefix(prefix:String): void
+ getTitols_autor(autor:String): ArrayList<String>
+ nombreAutors(): int
+ printAutors(): void
+ printTitols_autor(autor:String): void
+ getTitiols(autor:String): Object[]
+ getAutorsP(prefix:String): Object[]
1
Cjt_Autors
+ cjt_autors: ArrayList<String>
+ A_titol: HashMap<String, ArrayList<String>>
+ Cjt_autors()
+ add_autor_titiol(autor: String, titol: String): void
+ delete_titol_dautor(autor: String, titol: String): void
+ existeix_titol(titol:String): boolean
+ existeix_autor(autor:String): boolean
+ autor_conte_titol(autor: String, titol: String): boolean
+ delete_autor(autor:String): void
+ get_autors_prefix(prefix:String): ArrayList<String>
+ nombre_autors(): int
+ get_titols_autor(autor:String): ArrayList<String>
+ printAutors(): void
+ printTitolsAutor(autor:String): void


GestorDocuments
+ TITLE_KEY: String
+ AUTHOR_KEY: String
+ CONTENT_KEY: String
- priorityMap: HashMap<String,Integer>
- bracketPairs: HashMap<Charachter,Character>
- documents: ArrayList<Document>
- hashAuthorTitle: HashMap<String, ArrayList<Integer>>
- fraseReference: ArrayList<Pair<String, String>>
+ GestorDocuments()
+ addDocument(title:String, author:String, content:String): String[]
+ getDocuments(): ArrayList<HashMap<String,String>>
+ getDocument(i:int): HashMap<String,String>
+ getAuthorTitles(autor:String): void
+ getContent(i:int): String
+ getContent(author:String, title:String): String
+replaceContent(i:int, content:String): String[]
+replaceContent(title:String, author:String, content:String): String[]
+ printDocument(i:int): void
+ printDocumentList(): void
+ getDocumentList(): Object[]
+ deleteDocument(i:int): boolean
+ deleteDocument(title:String, author:String): boolean
+ getDocumentSize(): int
+ existsDocument(title:String, author:String): boolean
+ existsDocument(i:int): boolean
+ evaluateExpresion(e:String): ArrayList<String>
- isBalanced(str:String): boolean
- getDocument(title:String, author:String): Document
- creteContingut(content:String): Contingut
- getContentWords(content:String): String[]
- updateHashAuthor(): void
- updateReferences(i:int):void
Document
+ contingut: Contingut
+ title: Frase
+ author: Frase
+ Document(title:String, author:String, content:String)
+ getContingut(): String
+ getContingutF(): ArrayList<Frase>
+ getTitle(): String
+ getAuthor(): String
+ setContent(contingut:Contingut)
+ printInfo(): void
+ getInfo(): String
1
1
titol
autor
1
1
1
Contingut
- text: ArrayList<Frase>
+ Contingut()
+ Contingut(text:ArrayList<Frase>)
+ Contingut(text: String)
+ getText(): ArrayList<Frase>
+ setText(text: ArrayList<Frase>): void
+ getString(): String
+ printInfo(): void
Frase
+ paraules: ArrayList<String>
+ Frase()
+ Frase(text:String)
+ getString(): String
+ printInfo(): void


GestorMatriu
+ M: Matriu
+ funcionls: ArrayList<String>
+ GestorMatriu(paraules_func:ArrayList<String>)
+ add_pFuncionals(paraules_func:ArrayList<String>): void
+ addDocument_buit(titol:String, autor:String): boolean
+ addParaula(titol:String, autor:String, paraula:String): boolean
+ addContingut(titol:String, autor:String, paraula:String[]): void
+ replaceContingut(titol:String, autor:String, paraula:String[]): void
+ existeixDocument(titol:String, autor:String): boolean
+ deleteDoc(titol:String, autor:String): boolean
+ deleteParaula(titol:String, autor:String, paraula:String): boolean
+ getDocs_similars_Cos(titol_consultat:String, autor:String): PriorityQueue<Pair<String,Double>>
+ getDocs_similars_IDFJ(titol_consultat:String, autor:String): PriorityQueue<Pair<String,Double>>
+ printDocs_similars_Cos(titol_consultat:String, autor:String, N:int): boolean
+ printDocs_similars_IDFJ(titol_consultat:String, autor:String, N:int): boolean
+ getDocsSimilars_Cosinus(titol_consultat:String, autor:String, N:int): Object[]
+ getDocsSimilars_IDFJ(titol_consultat:String, autor:String, N:int): Object[]
1
Matriu
M: HashMap<String, HashMap<String, Integer>>
MatriuTF_IDF: HashMap<String, HashMap<String, Double>>
IDFJ: TreeMap<String, TreeSet<String>>
+ Matriu()
+ existeix_Par(paraula:String): boolean
+ add_Doc_buit(titol:String, autor:String): void
+ add_Parauls(titol:Strig, autor:String, paraula:String): void
+ delete_Doc(titol:String, autor:String): void
+ delete_Par(titol:Strig, autor:String, paraula:String): void
+ Doc_te_Par(titol:Strig, autor:String, paraula:String): boolean
+ Docs_Similars_Cos(titol:String, autor:String): PriorityQueue<Pair <String, Double>>
+ Docs_Similars_IDFJ(titol:String, autor:String): PriorityQueue<Pair <String, Double>>
+ calcularIDFJ(document:String): void
+ printIDFJ(): void

Understanding Classes:

Name: Archive
Description: File created by the user to import or export text documents.
Attributes: None attributes.

Name: Binary

Description: Class that allows the user to obtain and keep a class intàncies in binary format.

Attributes: None Attributes

Name: Cjt_Autors
Description: Set of authors and titles created by the user to add, delete or obtain authors and titles.
Atribut1:
-Name: cjt_autors
-Description: Contains the full name of the authors. Each author is unique.
-Estatic: No
Atribut2:
-Name: A_titol
-Description: For each author containing a collection of titles.
-Estatic: No

Name: Content
Description: content created by the user to modify and get content.
Atribut1:
-Name: text
-Description: Contains a set of phrases.
-Estatic: Yes
Atribut2:
-Name: stringtext
-Description: Saves the contents ams line breaks
-Estatic: Yes

Name: Document
Description: document created by the user to modify and obtain documents
Atribut1:
-Name: content
-Description: Document Content
-Estatic: No
Atribut2:
-Name: title
-Description: Document Title
-Estatic: No
Atribut3:
-Name: author
-Description: Author of the document
-Estatic: No

Name: Phrase
Description: phrase created by the user to modify and get sentences
attribute:
-Name: words
-Description: Words of a sentence.
'I'm Not

Name: Matrix
Description: Matrix created by the user to store a sparse matrix with nomñes
the nonzero value
Atribut1:
-Name: M
-Description: Contains titles and authors for any word contains appearances.
-Estatic: No
Atribut2:
-Name: MatriuTF_IDF
-Description Matrix has weights IDF
-Estatic: No
Atribut3:
-Name: IDFJ
-Description Matrix as a key word and content are documents that have that word.
-Estatic: No

Name: GestorArchivo
Description: File Manager to manage user-created class File Attribute:
-Name: arch
-Description Archive a File Manager
-Estatic: Yes

Name: GestorCjt_Autors
Description: Manager group of authors from user to manage the class Cjt_Autors
Attribute:
-Name: C
-Description: Set of authors of a manager set of authors
-Estatic: No

Name: GestorDocuments
Description: Document Manager to manage user-created class Atribut1 Document:
-Name: TITLE_KEY
-Description: Password Hashmap with a content of a title
-Estátic: No
Atribut2:
-Name: AUTHOR_KEY
-Description: Password Hashmap with a content of an author
-Estátic: No
Atribut3:

-Name: CONTENT_KEY
-Description: Password Hashmap one containing a content
-Estátic: No

Atribut4:
-Name: documents
-Description: Set Document Manager docuemnts
-Estátic: No

Atribut5:
-Name: hashAuthorTitle
-Description: Contains authors and titles of documents
-Estátic: No

Atribut6:
-Name: priorityMap
-Description: Contains the characters "(", "|", "!", "&"
-Estátic: Yes

Atribut7:
-Name: bracketPairs
-Description: Contains a pair of each type of brackets
-Estátic: Yes

Atribut8:
-Name: fraseReference
-Description: Keep the sentences of all documents
-Estátic: Yes

Name: GestorMatriu
Description: Manager of a matrix created by the user to manage class Matrix

Atribut1:
-Name: M
-Description: Array of arrays Manager
-Estátic: No

Atribut2:
-Name: functional
-Description: Contains functional words
-Estátic: No

Name: CtrlDomini
Description: Class containing a set of managers that allows the user to add, delete, obtain, modify, import, export documents. It can also create a list of authors, titles and documents.
Atribut1:
-Name: gestorDocuments
-Description: Manage documents
-Estàtic: Yes
Atribut2:
-Name: gestorCjAutors
-Description manages a group of authors
-Estàtic: Yes
Atribut3:
-Name: gestorMatriu
-Description manages comparing documents
-Estàtic: Yes
Atribut4:
-Name: gestorFitxers
-Description manages the import and export files
-Estàtic: Yes
Atribut5:
-Name: pFuncionals
-Description: Contains functional words
-Estàtic: Yes

Name: Pres_DelDoc
Description: interactive menu allows the user to delete one or more documents.

Name: Pres_AddDoc
Description: interactive menu allows the user to add one or more documents.

Name: Pres_DS
Description: interactive menu allows the user to list documents by calculating the cosine method or the method by calculating TF-IDFJ.

Name: Pres_EB
Description: interactive menu allows the user to list the documents that meet the Boolean expression.

Name: Pres_ExpDoc
Description: interactive menu that allows the user to export one or more documents.

Name: Pres_ImpDoc
Description: interactive menu allows the user to import one or more documents.

Name: Pres_ImpPF
Description: interactive menu allows the user to import a set of functional words.

Name: Pres_Llista
Description: interactive menu allows the user to display different content depending on the lists you have wanted list.

Name: Pres_ModDoc
Description: interactive menu allows the user to modify one or more documents.

Name: Pres_TA
Description: interactive menu allows the user to list the titles of an author.

Name: Pres_AP
Description: Interactive Menu allows the user to list authors as prefix.

Name: Pres_ConDoc
Description: Interactive Menu allows the user to view the contents of a document.

CtrlPresPrincipal Name:

Description: interactive menu allows the user to load and save sessions and add, delete, .obtenir, modify, import, export documents. It can also create a list of authors, titles and documents.

Atribut1:

-Name: openAction

-Description: Used to log

-Estàtic: Yes

Atribut2:

-Name: saveAction

-Description: Used to store a session

-Estàtic: Yes

Atribut3:

-Name: addDocAction

-Description: Used to add a document

-Estàtic: Yes

Atribut4:

-Name: delDocAction

-Description: Used to remove a document

-Estàtic: Yes

Atribut5:

-Name: modDocAction

-Description: Used moidificar document

-Estàtic: Yes

Atribut6:

-Name: impDocAction

-Description: Used to import a document

-Estàtic: Yes

Atribut7:

-Name: impPFAction

-Description: Used to import functional words

-Estàtic: Yes

Atribut8:

-Name: expDocAction

-Description: Used ecportar document

-Estàtic: Yes

Atribut9:

-Name: conDocAction

-Description: Used to view a document

-Estàtic: Yes

Atribut10:

-Name: cercaTAAction

-Description: Used to log

-Estàtic: Yes

Atribut11:

-Name: openAction

-Description: Used illistar titles of an author

-Estàtic: Yes

Atribut12:

-Name: openAction

-Description: Used to log

-Estàtic: Yes

Atribut13:

-Name: cercaAPAction

-Description: Used iconsultar authors prefix

-Estàtic: Yes

Atribut14:

-Name: cercaDocsAction

-Description: Used to list the documents that are in the system

-Estàtic: Yes

Atribut15:

-Name: cercaDSAction

-Description: Used to list similar documents

-Estàtic: Yes

Atribut16:

-Name: cercaEBAction

-Description: Used to list documents that meet a Boolean expression

-Estàtic: Yes

3. Description of data structures and algorithms

Data Structures:

class Cjt_Autors

The class contains two structures Cjt_Autors the first is a list of the names of authors in the system sorted alphabetically. The second contains a HashMap where each key corresponds to a copyright and the value of each key corresponds to a list, represented by an ArrayList that contains the titles of the works that the author has written in alphabetical order.

First class

The Array class contains a sparse matrix, ie a matrix where each row contains the same number of columns. Each row corresponds to a document identified by its title and author, and within each row are non-functional set of words of content and the number of appearances. To implement it using a HashMap where the keys, that would be the rows of the matrix are the titles and authors indicate the document. Each key contains another HashMap key words which are not functional content of that document, each word contains an integer corresponding to the number of occurrences of that word in that content.

Pair class

The Pair class is the representation of the combination of two values, the first is the identifier of a document, ie its title and its author, and the second a real number that corresponds to the degree of resemblance with another document.

class phrase

The class contains a single phrase structure is a list containing the words of the sentence. It is an ArrayList of string can also include punctuation as the phrase is separated by spaces disregarding punctuation, and taken into account before.

class content

Content contains a single class structure is a list of the phrases it contains. This list is separated through punctuation, to each point is considered a sentence. ArrayList is represented in the form of a phrase.

Document class

The class document contains three structures, two types of sentence represents a document title and the other author of the work. Finally structure contains a content type that stores the contents of the document according to the specifications of the class content.

class ExpresionTree

ExpresionTree class contains two structures, a type HashMap that contains a list of priority operations and a Boolean expression that represents the root node of the tree represented by the expression.

Node class

The class Node contains three structures, two of the same type Node representing the left and right children (if any) and another variable that contains the value of the node.

Explanation of algorithms:

Qualifications of the works that an author has written:

For titles that a particular author has written is returned alphabetically sorted list of titles from the HashMap key that corresponds to the name of the author.

Getting the authors according to their prefix:

For authors from a prefix searches with dichotomous prefix by the user on the list of authors in alphabetical order, once known position where the code is checked with respect to the name list author prefix that begins and progresses until there is no one that does not meet or has reached the end of the list.

Obtaining content given its title and author

For the content of a document because its title and author, first author found that exists in the HashMap manager, where are all the authors as a key and value is a list of positions where the author has a document. From this list covers looking title that matches the given parameter and then returns the content of the document.

Qualifications that meet a Boolean expression:

For the list of documents that meet a Boolean expression the first thing to consider is that the expression is given to the type of inordre a tree, and to build the tree that represents this expression must become the expression post-order. You should also check that the expression is well parenthesized, and therefore it is found that for every parenthesis or brace that there is a fence by a battery. Once converted, you must build the tree that represents the expression where such operations are contained as represented by a node. From the tree, you spend all the sentences of all the contents of the program, and we are eliminating a whole as not complying with the different sub-parts of the speech, only to finally get the resulting set of sentences that fulfilled.

Calculation of similarities between documents Cosine method:

To calculate the similarities between the two documents are represented as vectors set of words that are not functional so as follows:

Document 1 = (1 pes_del_terme weight _del_terme 2, ..., n pes_del_terme)
Document 2 = (1 pes_del_terme weight _del_terme 2, ..., n pes_del_terme)

.
.
.

documentN = (1 pes_del_terme weight _del_terme 2, ..., n pes_del_terme)

document_consulta = (1 pes_del_terme weight _del_terme 2, ..., n pes_del_terme)

On each weighing out corresponds to the number of appearances each word that has no functional content of the document. To determine the similarity between two documents is calculated by the cosine of the angle formed by applying the law of cosines, ie with the following formula:

$$\text{body}(\vec{v}, \vec{w}) = \frac{\vec{v} \times \vec{w}}{\overline{||\vec{v}|| \cdot ||\vec{w}||}}$$

where $\vec{v}$ and $\vec{w}$ represent two documents, $\vec{v} \times \vec{w}$ is the dot product of two vectors $||\vec{v}|| \cdot ||\vec{w}||$is the product of its modules. The result gives a number between 0 and 1 represents the degree of similarity that has a document with one another, the closer to 1 are more similar documents and the closer to 0 more different.

Calculation of similarities between documents TF-IDF method:

To calculate the similarities between the two documents with the method TF-IDF is represented as a vector set of words is not functional with the same structure as the calculation of similarities method cosine but with a different interpretation of Pessoa terms are calculated with the following formula:

$$\log \left(\frac{d}{dfj} \right)$$

where d is the total number of documents and dfj is the number of documents that contain that term which is calculated its weight.

To calculate the similarity between two documents is for the scalar product ($\vec{v} \times \vec{w}$) And the result gives a number between 0 and 1 is the degree of similarity that have the two documents.

4. Implemented List of Classes

class	Author
Archive	John Bryan Game Glices
binary	John Bryan Game Glices
Cjt_Autors	Albert Ruiz Lombarte
content	Aleix Rosello Rosiñol
CtrlDomini	Albert Ruiz Lombarte
CtrlPresPrincipal	Albert Ruiz Lombarte
document	Aleix Rosello Rosiñol
ExpresionTree	Aleix Rosello Rosiñol
phrase	Aleix Rosello Rosiñol
GestorArchivo	John Bryan Game Glices
GestorCjt_Autors	Albert Ruiz Lombarte
GestorDocuments	Aleix Rosello Rosiñol
GestorMatriu	Albert Ruiz Lombarte
matrix	Albert Ruiz Lombarte
node	Aleix Rosello Rosiñol
Pair	Albert Ruiz Lombarte
PairComparator	Albert Ruiz Lombarte
Pres_AddDoc	Albert Ruiz Lombarte
Pres_AP	Albert Ruiz Lombarte
Pres_ConDoc	Albert Ruiz Lombarte
Pres_DelDoc	Albert Ruiz Lombarte
Pres_DS	Albert Ruiz Lombarte
Pres_EB	Albert Ruiz Lombarte
Pres_ExpDoc	Albert Ruiz Lombarte
Pres_ImpDoc	Albert Ruiz Lombarte
Pres_ImpPF	Albert Ruiz Lombarte
Pres_Llista	Albert Ruiz Lombarte
Pres_ModDoc	Albert Ruiz Lombarte
Pres_TA	Albert Ruiz Lombarte

5. Relationship of external libraries used

The only external libraries we used are:

1. java.util. *

Concremanent: Java.utils.ArrayList, Java.utils.Collections, and Java.utils.Comparators Java.utils.PriorityQueue

Used to manage lists ArrayList called for the use of search algorithms and dichotomous arrangement in addition to comparators arranged in priority queues.

2. java.io. *

Specifically: Java.io.BufferedReader, Java.io.BufferedWriter, java.io.IOException, Java.io.InputStream

Used to manage data input and program output via either text files or binary files, as well as exceptions that can be generated.

3. javax.swing. *

Specifically: javax.swing.AbstractAction, javax.swing.Action, javax.swing.ImageIcon, javax.swing.JdesktopPane, javax.swing.JfileChooser, javax.swing.Jframe, javax.swing.JinternalFrame, avax.swing.Jmenu, javax.swing.JmenuBar, javax.swing.JoptionPane.

Used to create and manage components of the presentation whether panels, buttons, text boxes, lists, file browser, etc ...

6. List of features implemented

1. Management session:

Description: Allows the user to navigate the functions related to the management of its session.

behavior: The user can select from among the options management functionality of the session you want to use.

2. Open Session:

Description: Allows the user to open an information session that had previously saved.

behavior: The user selects the binary file browser files in the directory you had saved.

3. Save Session:

Description: Allows the user to save information from the current session.

behavior: The user indicates the directory name and file browser session you want to save.

4.Sortir:

Description: Allows the user to terminate the program.

behavior: The system ends the process was running.

5. Data management:

Description: Allows the user to navigate through the functions related to the management of data that has the system.

behavior: The user can select from among the options management functionality you want to use.

6. Add Document:

Description: Allows the user to add a new document to the system.

behavior: The user indicates the title, author and content of the new document we want to add, if the author does not exist with the added new work, if there is added to the author who wrote the work, and if
the author has already written a work with this title to indicate user error.

7. Change Document:

Description: Allows the user to modify any document in the system.

behavior: The user indicates the title and author of the document you want to change its content, if the author does not exist or has not written a work with this title indicate an error to the user, but can modify content successfully.

8. Remove Document:

Description: Allows the user to permanently delete any document in the system.

behavior: The user indicates the title and author of the document you want to delete the system does not exist or if the author has not written a work with this title will indicate to the user an error, but the document will be cleared from the system.

9. Functional Subject Matter:

Description: Allows the user to import a file containing a functional words that are not taken into account when calculating the similarities between documents.

behavior: The user indicates the file browser with the absolute path and filename you want to import, or if the path name does not indicate a valid exception. If correct, functional words added to the list of words to ignore when calculating similarities between documents.

10. Import Document:

Description: Allows the user to import a file containing information about a document.

behavior: The user indicates the title, author, and file browser with absolute path and file name with the content you want to import, or if the path name is invalid indicate an error. If it is correct.
If the author added there with his new work, if there is added to the author who wrote the work, and if the author has already written a work with this title to indicate the user error.

11. Export Document:

Description: Allows the user to export a file containing information a document.

behavior: The user indicates the absolute path and filename you want to export the file browser, or if the path name does not indicate a valid exception. If correct the content of that document will be saved in the text file.

12. See details:

Description: Allows the user to navigate through the functions related queries the system data.

behavior: The user can select from the options query functionality to use.

13. Consult a Title Author:

Description: Allows the user to see the titles of the works he has written a particular author.

behavior: The user indicates the name of the author which wants to see the titles of his works, if the author is there to tell the user with an error.

14. Ask for Authors prefix:

Description: Allows the user to consult the authors of the system through a prefix.

behavior: The user indicates the prefix, which may be empty, and it will show the names of authors that start with this prefix.

15. Check List Documents:

Description: Allows the user to view the list of documents in the system.

behavior: The user is shown a list of all the documents identified by author and title in the system.

16. Consult Content:

Description: Allows the user to view the contents of a particular document.

behavior: The user indicates the title and author of the document you want to view your content, if the author does not exist or has not written a work with this title will tell the user with an error.

17. See similar documents:

Description: Allows the user to see the subset of documents similar to a given.

behavior: The user indicates the title and author of the document you want to view the subset of similar documents, if the author does not exist or has not written a work with this title will tell the user with an error . Then it will ask the user to the method of calculating similarities you want to use and a natural number k and k documents will be displayed,
title, author and degree of similarity, ordered more like less similar to what he said.

18. Documents See Boolean expression:

Description: Allows the user to see the subset of documents the content of which meets a specific Boolean expression.

behavior: The user indicates a Boolean expression and it will show the titles and authors of works which fulfill the content of the speech. But if you are not in the correct format will indicate to the user with an error.

19. Help

Description: Allows the user to browse features help.

behavior: The user can select from among the options functionality that helps to use.

20. User:

Description: Allows the user to consult the user manual of the program

behavior: Open a PDF file viewer for the user to see the manual.

21. About:

Description: Allows the user to check the authorship of the software.

behavior: This opens a warning message with information on the authorship of the program.

7. User

User

Programming Project 22 Cluster 6.2 Subgroup

Management session:

Navigate to the features related to the management of the session that contains all changes to data stored so far. There are the following features that are presented below:

Open session:

Opens with the binary data from a saved session anteriormemt, to select the binary search in the file browser Once found, click on File and then Open.

Save Session:

Can save in a binary data changes that have occurred in the current session. To create the file you must select the directory with the file browser in which you will create the binary data stored. Once selected directory, called the file and click the Save button.

To go out:

Will close the program that was executant if unsaved changes in data functionality Save Session will be lost irretrievably.

Data management:

Navigate to the features related to the management of data that has the system. There are the following functions for managing data:

Add Document:

Adds a document, entering the title, author and content in different fields blank, you can not add any document containing any of the fields empty.

Clear Document:

Erase the document data indicating which holds the title and author.

Modify Document:
Modify the continent in a document. To do this you must indicate the title and author of the document to be modified, press the button to show the contents indicate the current contents of the document to be modified, once modified to press the button to modify the document to save the changes.

Import Document:
Allows importing the content of a document that is in a text file to make it possible to indicate the title and the name of its author, then you need to select the text file with the file browser and select Open. Once this is done the system will have a new document with the title and author and content as indicated in the text containing the selected file.

Functional Subject Matter:
Allows importing a file with new functional words are not taken into account in the calculation of similarities between documents. To do so, select the text file with the file browser and click the Open button. Once this is done the system interpertarà each line of text file as a functional word is not taken into account in the calculation of similarities between documents.

Export Document:
To export the contents of a document to make it possible to indicate the title and author of the document and indicate the name of the file you exported the contents of the document and press the Save button.

Queries system data:
Navigate to features related queries system data. There are the following features to search data in the system:

Consult Content:
View the contents of a document indicating its title and its author.
To do this you must indicate the title and author and press the button to display content.

Browse titles author:
View the titles of documents that a particular author has written indicating the name of the system. To do indicate the author's name and titles listed Permission button.

List authors prefix:
View author names that begin with a certain prefix by the user. To do this you must indicate the prefix to be empty and press the button to list authors.

Browsing documents:
View the titles and authors of all documents that have the system.

Documets similar list:
View the N documents similar to a given two methods for calculating similarities content (Cosine and TF-IDFJ). To do so the user should indicate the title and author of the document you want to look like her, the number of documents you want to see, select the method of calculating similarities want to use and press list of similar documents. Then he displayed the N documents similar to the given order of similarity and the degree of similarity indicated by a number between 0 and 1.

Browsing documents Boolean expression:
You can view documents that meet the Boolean expression indicated. To do well you must indicate the parenthesized Boolean expression and press the button to search for documents.

Help

Lets browse functionality helps the user and program information, there are the following features related to user assistance and information:

User:
View the PDF document with the functionality of the program.

About:
View authoring software.

8. Games testing complete

test 1
Description: Play the following funcionailtats test test in the following order:
1. Import Document 2. Consult Documents, Documents 3. Check similarities between the method of calculation of TF-IDFJ.
The test will consist of the following:
1. Import documetns have the following content: D1: the
Danube pasa por Vienna, its color is blue
D2: The flow of a river in winter asciende
D3: the river Rhine and the Danube Tienen mucho flow
D4: If a river is navigable, it porque tiene mucho flow
Q: which is the flow of the Danube
2. Check that the documents actually part of the system
3. Check the similarity of Q documents with everyone else to see if that is really the calculation below is correct:

We are following matrix Pessoa where the number of occurrences of each word:

	Río	Danube	Vienna	color	azul	caudal	winter	Rhine	navigable
D1	1	1	1	1	1	0	0	0	0
D2	1	0	0	0	0	1	1	0	0
D3	2	1	0	0	0	1	0	1	0
D4	1	0	0	0	0	1	0	0	1

Calculation of inverse frequencies:

IDF (river) = log (4/4) = log (1) = 0
IDF (Danube) = log (4/2) = log 2 = 0301 IDF (Vienna) = log (4/1) = log 4 = 0602 IDF (color) = log (4/1) = log 4 = 0602
IDF (azul) = log (4/1) = log 4 = 0602
IDF (flow) = log (4/3) = log = 0. 1:33 124 IDF (winter) = log (4/1) = log 4 = 0602 IDF (Rhine) = log (4/1) = log 4 = 0602
IDF (waterway) = log (4/1) = log 4 = 0602

It is the TF-IDF following matrix:

	Río	Danube	Vienna	color	azul	Caudal	winter	Rhine	navigable
D1	0	0301	0602	0602	0602	0	0	0	0
D2	0	0	0	0	0	0124	0602	0	0
D3	0	0301	0	0	0	0124	0	0602	0
D4	0	0	0	0	0	0124	0	0	0602
Q	0	0301	0	0	0	0124	0	0	0

Calculation of similarities (scalar product of vectors):

Sim (D1, Q) = 0 * 0 * 0301 + 0301 + 0602 + 0602 * 0 * 0 * 0 + 0 + 0 602 0 *. 124+ 0 * 0 + 0 * 0 + 0 * 0 = 0.090601 ...
Sim (D2, Q) = 0 + 0 * 0 * 0 * 0301 + 0 + 0 + 0 * 0 * 0 * 0 + 0. 124. 124 + 0,602 * 0 + 0 * 0 + 0 * 0 = 0.0153376
Sim (D3, Q) = 0 + 0 * 0301 * 0301 * 0 + 0 + 0 + 0 * 0 * 0 * 0 + 0. 124. 124 + 0 * 0 + 0,602 * 0 + 0 * 0 = 0.105977
Sim (D4, Q) = 0 * 0 + 0 * 0,301 + 0 * 0 + 0 * 0 + 0 * 0 + 0,124 * 0,124 + 0 * 0 + 0 * 0 + 0,602 * 0 = 0.015337

Then we see as a result of the similtud Q system with the following documents ordered list of most similar unless:

D3 (0.105977)
D1 (0.090601)
D2 (0.015337)
D4 (0.015337)

Objectives: The main objective is to verify the proper operation of the method of calculation of TF-IDF to use the functionality necessary for using this core functionality to be tested.

Admission:

Data Management> Import Document
Title: D1
Author: nom_autor
File Browser: EXE / data / Prova1 / D1.txt
Open
Title: D2
Author: nom_autor
File Browser: EXE / data / Prova1 / D2.txt
Open
Title: D3
Author: nom_autor
File Browser: EXE / data / Prova1 / D3.txt
Open
Title: D4
Author: nom_autor
File Browser: EXE / data / Prova1 / D4.txt
Open
Title: Q
Author: nom_autor
File Browser: EXE / data / Prova1 / Q.txt
Open
Close
Search> Browse Documents

Way out:
D1 - nom_autor
D2 - nom_autor
D3 - nom_autor
D4 - nom_autor
Q - nom_autor
Admission:
Close
Search> List similar documents
Title: Q
Author: nom_autor
What calculation method you want to use? TF-calculation
method IDFJ How many documents do you want? 4
Way out:
1. D3 - nom_autor: 0.105977
2. D1 - nom_autor: 0.090601
3. D2 - nom_autor: 0.015337
4. D4 - nom_autor: 0.015337

Test 2:
Description: Game Test complete the following test funcionailtats in the following order:
1. Import Document, Documents List 2., 3. Consult with prefix Author, Exit 4. 5. 6. Open Session Save Session 7. Documents List, 8. Consult content.

Admission:
Data Management> Import Document
Title: The art of (not) shut
Author: Vera Esther
File Browser: EXE / data / Prova2 / article1.txt
Title: Perceptions and Realities
Author: Carme Colomina
File Browser: EXE / data / Prova2 / article2.txt
Title: Sometimes we go out on, Sebastian
Author: Carles Capdevila
File Browser: EXE / data / Prova2 / article3.txt
Title: The End of the World
Author: Fernando Trias de Bes
File Browser: EXE / data / Prova2 / article4.txt
Title: Be sure to know
Author: Joaquim Coello
File Browser: EXE / data / Prova2 / article5.txt
Search> Browse Documents
Way out:
The art of (not) shut - Esther Vera
Perceptions and Realities - Carme Colomina
Sometimes we go out on, Sebastian - Carlos Capdevila
The end of the world - Fernando Trias de Bes
Be sure to know - Joaquim Coello

Admission:
Search> Browse Author prefix
Prefix: Car
List authors prefix
Way out:
Carles Capdevila
Carme Colomina

Admission:
Log> Save Log
Browser fitxres: EXE / data / Prova2
File Name: prova2
Session> Exit
Open the program again
Log> Open Session
Browser fitxres: EXE / data / Prova2 / prova2
Search> List documents
Way out:
The art of (not) shut - Esther Vera
Perceptions and Realities - Carme Colomina
Sometimes we go out on, Sebastian - Carlos
Capdevila The end of the world - Fernando Trias de
Bes
Be sure to know - Joaquim Coello
Arrival:
Search> Browse Content
Title: Perceptions and Realities
Author: Carme Colomina
Search content
Way out:
(Contents of the document)

test 3

Description: Play test used to test the functionality of query Boolean expression.
The test consists of the following:
1. Import the various items that are in the folder EXE / data / Prova2 /
2- We Boolean query follows:
((Silence & Culture) | ("early December" &! Silence))
3. Check that docuemento to comply from the content

IMPORTANT:
Data Management> Import document
Enter data
Title: t0
Author: a0
so on ...

Then we consult with:
Search> Browse documents Boolean expression

Enter the following expression:
((Culture) | ("early December" &! Silence))

We see that the output is:
t0 - a0
t1 - a1

They are actually documents that meet the Boolean expression.